

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**МЕТОДИЧНІ ВКАЗІВКИ**

**до лабораторної роботи**

**«Створення простих запитів. Команда Select мови SQL»**

з курсу «Проектування баз даних» для студентів спеціальності

122 Комп'ютерні науки, з курсу «Бази даних» спеціальності

186 Видавництво та поліграфія та з курсу «Проектування  
інформаційних систем» спеціальності 124 Системний аналіз

Затверджено

редакційно-видавничою

радою університету,

протокол № 1 від 30.01.2018 року

Харків  
НТУ «ХПІ»

2018

Методичні вказівки до лабораторної роботи «Створення простих запитів. Команда Select мови SQL» з курсу «Проектування баз даних» для студентів спеціальності 122 Комп'ютерні науки та інформаційні технології, з курсу «Бази даних» спеціальності 186 Видавництво та поліграфічна справа та з курсу «Проектування інформаційних систем» спеціальності 124 Системний аналіз / уклад.: Л. Б. Кашеєв, С. В. Коваленко, О. С. Куценко – Харків : НТУ «ХПІ», 2018. – 16 с.

Укладачі:           Л. Б. Кашеєв,  
                              С. В. Коваленко,  
                              О. С. Куценко

Рецензент Л. М. Любчик

Кафедра системного аналізу та інформаційно-аналітичних технологій

## ВСТУП

Метою даної лабораторної роботи є ознайомлення з можливостями команди SELECT мови SQL. Дана команда є частиною мови DML та виконує вибірку даних з однієї або декількох таблиць одночасно. В даній роботі потрібно виконувати дії тільки з однією таблицею. Будуть розглянуті основні частини команди SELECT – список вибірки та розділи FROM, WHERE, ORDER BY.

## 1. ТЕОРЕТИЧНІ ОСНОВИ МОВИ SQL

Мова SQL (Structured Query Language) була розроблена в 1974 році в ході роботи над реляційною СУБД System R. У період становлення технологій баз даних SQL був не єдиним і, можливо, не найкращим цих слів реляційних баз даних, але з часом став стандартом в цій області. Комерційні продукти далеко не відразу забезпечили властивості вихідної версії SQL в повному обсязі.

Мова SQL призначена для маніпулювання даними в реляційних базах даних, визначення структури баз даних і для управління правами доступу до даних в багатокористувацькому середовищі. SQL є реляційно-повною мовою в тому сенсі, що вона забезпечує всі операції реляційної алгебри. Але разом з тим, в самих концепціях SQL є і деякі відступи від реляційної теорії.

В мову SQL в якості складових частин входять:

- мова маніпулювання даними (Data Manipulation Language, DML);
- мова визначення даних (Data Definition Language, DDL);
- мова керування даними (Data Control Language, DCL).

Підкреслимо, що це не окремі мови, а різні команди однієї мови. Такий поділ проведено тільки з точки зору різного функціонального призначення цих команд.

Мова **маніпулювання даними** використовується, як це випливає з його назви, для маніпулювання даними в таблицях баз даних. Вона складається з 4 основних команд:

SELECT (вибрати);

INSERT (вставити);  
UPDATE (оновити);  
DELETE (видалити).

Мова **визначення даних** використовується для створення та зміни структури бази даних і її складових частин – таблиць, індексів, уявлень (віртуальних таблиць), а також тригерів і збережених процедур. Основними його командами є:

CREATE DATABASE (створити базу даних);  
CREATE TABLE (створити таблицю);  
CREATE VIEW (створити віртуальну таблицю);  
CREATE INDEX (створити індекс);  
CREATE TRIGGER (створити тригер);  
CREATE PROCEDURE (створити збережену процедуру);  
ALTER DATABASE (модифікувати базу даних);  
ALTER TABLE (модифікувати таблицю);  
ALTER VIEW (модифікувати віртуальну таблицю);  
ALTER INDEX (модифікувати індекс);  
ALTER TRIGGER (модифікувати тригер);  
ALTER PROCEDURE (модифікувати збережену процедуру);  
DROP DATABASE (видалити базу даних);  
DROP TABLE (видалити таблицю);  
DROP VIEW (видалити віртуальну таблицю);  
DROP INDEX (видалити індекс);  
DROP TRIGGER (видалити тригер);  
DROP PROCEDURE (видалити збережену процедуру).

Мова **керування даними** використовується для управління правами доступу до даних і виконанням процедур в багатокористувацькому середовищі. Більш точно її можна назвати «мова управління доступом». Вона складається з двох основних команд:

GRANT (дати права);  
REVOKE (забрати права).

## 2. ВИХІДНІ ТАБЛИЦІ ДЛЯ ЗДІЙСНЕННЯ ВИБІРКИ ДАНИХ

В якості предметної області, що досліджується, зупинимося на добре відомій всім студентам інформації про успішність навчання в

рамках академічної групи. Ми вже маємо 3 таблиці – Студенти, Дисципліни та Іспити, що пов’язані постійними відносинами (рис. 2.1).

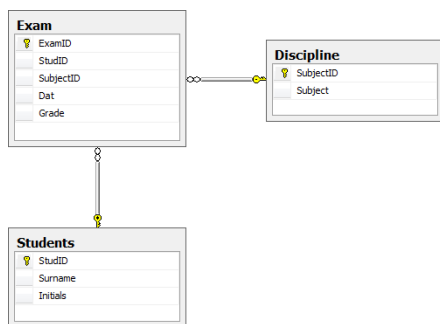


Рисунок 2.1 – Інфологічна модель розробленої бази даних

До наведених таблиць вже внесені дані, що будуть служити основою для побудови запитів. На рис. 2.2 наведено приклад цих таблиць.

SubjectID	Subject	StudID	Surname	Initials
1	Програмування	1	Борисов	Т.В.
2	Комп'ютерна графіка	2	Петров	В.Я.
3	Фізика	3	Охрименко	А.К.
4	Бази даних	4	Сидоренко	С.Ю.

ExamID	StudID	SubjectID	Dat	Grade
1	1	2	2017-01-21	94
2	2	2	2016-12-28	75
3	3	1	2017-01-15	94
4	1	3	2017-01-21	63
5	2	4	2016-12-28	82
6	4	4	2017-06-15	69

Рисунок 2.2 – Вміст розроблених таблиць

### 3. ПРОСТА ВИБІРКА ДАНИХ

Для простої вибірки даних використовується скорочений синтаксис оператора SELECT:

```
SELECT [ALL | DISTINCT] [TOP k [PERCENT]]
```

*Список\_вибірки*

```
FROM Ім'я_таблиці
```

*Список\_вибірки* задає стовпці, що включаються в підсумковий набір даних, *Ім'я\_таблиці* вказує звідки беруться записи.

**Завдання 1.** Вивести вміст таблиці Іспити:

```
SELECT * FROM Exam
```

Якщо цікавлять не всі стовпці, то їх потрібно перелічити:

```
SELECT ExamID, Dat, Grade  
FROM Exam
```

Ця дія часто носить назву вертикальної фільтрації – користувач бачить обмежену кількість стовпців.

**Завдання 2.** Вивести вміст таблиці Студенти, підписавши назви стовбців українською мовою та залишивши трьох перших студентів:

```
SELECT TOP 3 Surname as Прізвище, Initials as  
[Ініціали студента]  
FROM Students
```

Результат наведено на рис 3.1.

	Прізвище	Ініціали студента
1	Борисов	Т.В.
2	Петров	В.Я.
3	Охрименко	А.К.

Рисунок 3.1 – Результат завдання 2

Аналогічний результат можна отримати якщо вказати кількість студентів у відсотках від загального числа. При цьому округлення числа записів буде проводитися в більшу сторону – таким чином вказавши 51% від 4 значень, ми вже отримаємо три записи:

```
SELECT TOP 51 PERCENT Surname as Прізвище, Initials  
as [Ініціали студента]  
FROM Students
```

*Завдання 3.* Вивести із таблиці Іспити номер місяця та рік, коли студенти здавали іспити. Якщо будуть однакові набори значень, вони не повинні повторюватися:

```
SELECT DISTINCT month(Dat) as Місяць, year(Dat) as
Рік
FROM Exam
```

Опція DISTINCT потрібна для отримання набору унікальних (неповторюваних) записів. Результат наведено на рис 3.2.

Місяць	Рік
1	2017
6	2017
12	2016

Рисунок 3.2 – Результат завдання 3

*Завдання 4.* Вивести вміст таблиці Іспити, перетворивши оцінки з балів в шкалу ECTS:

```
SELECT ExamID, Dat, Grade, CASE
                                WHEN Grade < 34 THEN 'F'
                                WHEN Grade < 60 THEN 'Fx'
                                WHEN Grade < 67 THEN 'E'
                                WHEN Grade < 75 THEN 'D'
                                WHEN Grade < 82 THEN 'C'
                                WHEN Grade < 90 THEN 'B'
                                ELSE 'A'
                                END as 'ECTS'
FROM Exam
```

Результат запиту наведено на рис 3.3.

ExamID	Dat	Grade	ECTS
1	2017-01-21	94	A
2	2016-12-28	75	C
3	2017-01-15	94	A
4	2017-01-21	63	E
5	2016-12-28	82	B
6	2017-06-15	69	D

Рисунок 3.3 – Результат завдання 4

При побудові запитів часто представляють інтерес не всі записи, а тільки деякі, що відповідають певному критерію. Для вказівки критерію відбору записів в підсумковий набір використовується розділ WHERE. Умови відбору можуть складатися з однієї логічної умови або декількох, що поєднані логічними операціями and, or та not.

*Завдання 5.* Вивести вміст таблиці Іспити, залишивши тільки іспити, по яких було отримано більше 90 балів або менш 70:

```
SELECT ExamID, StudID, SubjectID, Dat, Grade
FROM Exam
WHERE Grade>90 or Grade<70
```

Результат запиту наведено на рис 3.4.

ExamID	StudID	SubjectID	Dat	Grade
1	1	2	2017-01-21	94
3	3	1	2017-01-15	94
4	1	3	2017-01-21	63
6	4	4	2017-06-15	69

Рисунок 3.4 – Результат завдання 5

Для обробки множини записів можна використовувати оператори in, between та like. Оператор in визначає набір значень, в яке дане значення може або не може бути включене.

*Завдання 6.* Вивести вміст таблиці Дисципліни, залишивши тільки назви «Програмування», «Проектування інформаційних систем», «Бази даних»:

```
SELECT Subject
FROM Discipline
WHERE Subject in ('Програмування', 'Проектування
інформаційних систем', 'Бази даних')
```

Результат запиту наведено на рис 3.5.

Subject
Програмування
Бази даних

Рисунок 3.5 – Результат завдання 6



Умова `between` повертає істинний результат, коли значення, що порівнюється, потрапляє в заданий діапазон, включаючи межі діапазону.

**Завдання 7.** Вивести вміст таблиці Іспити, залишивши тільки строки, в яких оцінка знаходиться від 75 до 90:

```
SELECT ExamID, StudID, SubjectID, Dat, Grade
FROM Exam
WHERE Grade between 75 and 90
```

Результат запиту наведено на рис 3.6.

ExamID	StudID	SubjectID	Dat	Grade
2	2	2	2016-12-28	75
5	2	4	2016-12-28	82

Рисунок 3.6 – Результат завдання 7

Умова `like` вимагає завдання шаблону, з яким порівнюється задане значення. Результат правдивий, якщо значення, що порівнюється, відповідає шаблону, і хибний в іншому випадку

**Завдання 8.** Вивести вміст таблиці Студенти, залишивши тільки строки, в яких прізвище починається на літеру від «Н» до «Р», містить літери «е» та «о», причому «е» іде раніше:

```
SELECT Surname, Initials FROM Students
WHERE Surname like '[H-P]%e%o%'
```

Результат запиту наведено на рис 3.7.

Surname	Initials
Петров	В.Я.
Охрименко	А.К.

Рисунок 3.7 – Результат завдання 8

Вказані оператори `in`, `between` та `like` можуть використовуватися з логічним оператором `not` – `not in`, `not between`, `not like`. Ці умови мають протилежний зміст.

Зручність сприйняття одержуваного набору даних може бути підвищена шляхом його сортування в зростаючому або спадному порядку. Сортування можливе по імені поля (навіть якщо воно і не вказано в списку вибірки), за псевдонімом поля або по позиції в списку вибірки, які вказуються в розділі ORDER BY. За замовчуванням сортування здійснюється за зростанням, що відповідає зарезервовані слова ASC, яке може опускатися, для сортування в порядку спадання вказується слово DESC.

*Завдання 9.* Вивести вміст таблиці Студенти, упорядкувавши строки проти абетки прізвищ:

```
SELECT Surname, Initials
FROM Students
ORDER BY Surname DESC
```

Результат запиту наведено на рис 3.8.

Surname	Initials
Сидоренко	С.Ю.
Петров	В.Я.
Охрименко	А.К.
Борисов	Т.В.

Рисунок 3.8 – Результат завдання 9

*Завдання 10.* Вивести вміст таблиці Іспити, упорядкувавши строки по зростанню місяця, якщо є однакові по зростанню дня здачі іспиту:

```
SELECT ExamID, StudID, SubjectID, Dat, Grade
FROM Exam
ORDER BY month(Dat), day(dat)
```

Результат запиту наведено на рис 3.9.

ExamID	StudID	SubjectID	Dat	Grade
3	3	1	2017-01-15	94
4	1	3	2017-01-21	63
1	1	2	2017-01-21	94
6	4	4	2017-06-15	69
2	2	2	2016-12-28	75
5	2	4	2016-12-28	82

Рисунок 3.9 – Результат завдання 10

#### 4. ВАРІАНТИ ЗАВДАНЬ НА ЛАБОРАТОРНУ РОБОТУ

##### *Варіант 1*

Для раніше створеної бази даних, що містить три зв'язані таблиці (перша таблиця містить поля – код і назва факультету; друга таблиця – код факультету, код спеціальності, найменування спеціальності; третя таблиця – номер залікової книжки, прізвище, ініціали, код спеціальності, стать, дата народження) виконати наступні запити:

1. Видати повну інформацію з третьої таблиці, підписавши назви колонок і перетворивши стать в словесний вид (чоловіча жіноча).

2. Видати з третьої таблиці стовпці: прізвище, стать, виключивши набори, що повторюються.

3. Вивести з третьої таблиці чоловіків, народжених після 1 січня 1999 року.

4. Вивести з третьої таблиці студентів з номерами залікових книжок, що закінчуються на непарну цифру.

5. Вивести з першої таблиці факультети, в найменуванні яких міститься слово «інформатик».

6. Вивести з третьої таблиці студентів з невизначеною датою народження.

7. Вивести з третьої таблиці студентів, упорядкувавши їх за алфавітом прізвищ, якщо є однакові, то за спаданням номерів залікових книжок.

8. Вивести з третьої таблиці студентів, упорядкувавши їх за зростанням дати народження в поточному році (за зростанням місяців і днів).

### *Варіант 2*

Для раніше створеної бази даних, що містить три зв'язані таблиці (перша таблиця містить поля – код і найменування фірм виробників автомобілів; друга таблиця – код фірми-виробника автомобілів, код моделі автомобіля, найменування моделі авто, довжина авто (мм), ширина авто (мм); третя таблиця – державний номер автомобіля, код моделі автомобіля, тип двигуна, колір) виконати наступні запити:

1. Видати повну інформацію з другої таблиці, підписавши назви колонок і додавши нову колонку – якщо площа, що зайнята над поверхнею землі, більше нуля, то вивести значення площі, в іншому випадку напис «невідомий параметр».

2. Видати з третьої таблиці стовпці: довжина, ширина моделі автомобіля, виключивши набори, що повторюються.

3. Вивести з третьої таблиці автомобілі з дизельним двигуном.

4. Вивести з другої таблиці автомобілі довжина яких знаходиться в інтервалі від 4200 до 4500 мм.

5. Вивести з другої таблиці моделі автомобілі, найменування яких починається на «В» і містить одну з цифр 0 або 2.

6. Вивести з третьої таблиці автомобілі з невизначеним кольором (значення NULL).

7. Вивести з третьої таблиці автомобілі, упорядкувавши їх за типами двигунів, якщо є однакові, то за алфавітом державних номерів.

8. Вивести з другої таблиці моделі автомобілів, упорядкувавши їх за зменшенням площі.

### *Варіант 3*

Для раніше створеної бази даних, що містить три зв'язані таблиці (перша таблиця містить поля – код кольору, колір; друга таблиця – код сімейства, найменування сімейства тварин (наприклад, котяті, бегемотові, ведмежі, слонові і т. п.); третя таблиця – порядковий номер, найменування тварини, кличка тварини, код сімейства, код кольору, вага, здатність плавати) виконати наступні запити:

1. Видати повну інформацію з третьої таблиці, підписавши назви стовпців і додавши новий – якщо вага до 1 кг, написати «маленька тварина», якщо від 1 до 10 кг – «середня тварина», від 10 до 50 кг – «велика тварина», понад 50 кг – «величезна тварина».

2. Видати з третьої таблиці стовпці: вага, вміння плавати,

виключивши набори, що повторюються.

3. Вивести з третьої таблиці тварин, які вміють плавати і вагою понад 100 кг.

4. Вивести з третьої таблиці тварин з найменуваннями «слон», «тигр», «лев».

5. Вивести з першої таблиці кольори, найменування яких починається на символ «з», «г» або «ж» або міститься в середині символ «е».

6. Вивести з третьої таблиці тварин з невизначеною кличкою (значення NULL).

7. Вивести з третьої таблиці тварин, упорядкувавши їх за зменшенням ваги, якщо є однакові, то за алфавітом найменувань тварини.

8. Вивести з третьої таблиці тварин, упорядкувавши їх за зростанням відмінності ваги від числа 100.

#### *Варіант 4*

Для раніше створеної бази даних, що містить три зв'язані таблиці (перша таблиця містить поля – код, прізвище абонента; друга таблиця – код, найменування вулиці; третя таблиця – номер телефону, код прізвища, ініціали, індекс, код вулиці, номер будинку, номер квартири, стать абонента) виконати наступні запити:

1. Видати повну інформацію з третьої таблиці, підписавши назви колонок і перетворивши стать в строкове представлення «чоловічий», «жіночий».

2. Видати з третьої таблиці стовпці: номер будинку, номер квартири, стать, усунувши набори, що повторюються.

3. Вивести з третьої таблиці людей чоловічої статі, які проживають в будинку «1а».

4. Вивести з третьої таблиці людей, у яких в номері телефону третя цифра 3, 7, або 8, п'ята цифра не 6, 3 і 7, а шоста цифра від 4 до 9.

5. Вивести з другої таблиці вулиці в найменування яких міститься слово «Героїв».

6. Вивести з третьої таблиці людей, які проживають в квартирах з невизначеним номером (значення NULL).

7. Вивести з третьої таблиці людей, упорядкувавши їх за зменшенням номера будинку, якщо є однакові, то по зростанню номера

квартири.

8. Вивести з третьої таблиці людей, упорядкувавши їх за зменшенням відмінності номера квартири від числа 13.

### *Варіант 5*

Для раніше створеної бази даних, що містить три зв'язані таблиці (перша таблиця містить поля – код, найменування видавництва; друга таблиця – код автора, прізвище автора, ініціали, дата народження; третя таблиця – код книги, код прізвища автора, найменування книги, код видавництва, кількість сторінок, жанр книги) виконати наступні запити:

1. Видати повну інформацію з третьої таблиці, підписавши назви стовпців і додавши новий стовпець, якщо кількість сторінок до 20 – «маленька книжка», від 20 до 50 – «середня книга», від 50 до 200 – «велика книга», понад 200 – «читати, не переріхувати».

2. Видати з третьої таблиці стовпці: кількість сторінок, жанр книги, усунувши набори, що повторюються.

3. Вивести з третьої таблиці наукові книги з кількістю сторінок більше 100.

4. Вивести з другої таблиці авторів, які народилися влітку.

5. Вивести з другої таблиці авторів прізвище яких починається на "Л", третій символ приголосна буква, слово закінчується на «ов».

6. Вивести з другої таблиці авторів з невизначеними ініціалами.

7. Вивести з третьої таблиці книги, упорядкувавши їх за жанром, всередині по спадаючій кількості сторінок.

8. Вивести з третьої таблиці книги, упорядкувавши їх за зменшенням відмінності кількості сторінок від числа 200.

### *Варіант 6*

Для раніше створеної бази даних, що містить три зв'язані таблиці (перша таблиця містить поля – код фільму, найменування фільму, тривалість; друга таблиця – код каналу, назва телевізійного каналу; третя таблиця – код, код найменування фільму, код каналу, час початку, час закінчення, обмеження за віком) виконати наступні запити:

1. Видати повну інформацію з третьої таблиці, підписавши назви стовпців і додавши новий стовпець – якщо фільм починається до 9

години – «ранок», від 9 до 14 – «день», від 14 до 20 – «вечір», пізніше 20 і до 9 – «ніч».

2. Видати з третьої таблиці стовпці: час закінчення, обмеження за віком, усунувши набори, що повторюються.

3. Вивести з третьої таблиці фільми, час початку яких від 9.00 до 15.00.

4. Вивести з другої таблиці канали, найменування яких починаються на голосну букву, а на третій позиції стоїть цифра.

5. Вивести з першої таблиці фільми, в найменуванні яких рівно 4 слова.

6. Вивести з третьої таблиці інформацію про фільми з невизначеним обмеженням за віком.

7. Вивести з третьої таблиці інформацію про фільми, упорядкувавши їх за зростанням часу початку, всередині по спадаючій часу закінчення.

8. Вивести з третьої таблиці інформацію про фільми, упорядкувавши їх за зростанням тривалості демонстрації (різниця між закінченням і початком).

### Контрольні запитання

Задана таблиця з назвою T

id	n	x	v	p
1	Слон	15	4900	1
2	Ведмідь	8	300	1
3	Жираф	10	1100	1
4	Вовк	3	60,8	1
5	Кіт	3	7,5	0
6	Лисиця	2	9,1	0

Що виведуть наступні запити?

1.

```
SELECT n, x, v
FROM T
WHERE x > 2 and v < 3000
```

2.

```
SELECT n, x, v, p
FROM T
WHERE x in (2,8,10)
```

3.

```
SELECT n, x, v, p
FROM T
WHERE x not between 7 and 10 and p='True'
```

4.

```
SELECT n, x, v, p
FROM T
WHERE n like '[BKJIC]____'
```

5.

```
SELECT n, x, v, p
FROM T
ORDER BY x DESC, v
```

6.

```
SELECT n, x, v, p
FROM T
WHERE v < 1000
ORDER BY p
```

## СПИСОК ЛІТЕРАТУРИ

1. Кренке Д. Теория и практика построения баз данных // Д. Кренке – СПб. : Питер, 2005. – 864 с.

2. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика // Т. Коннолли, К. Бегг – М. : Вильямс, 2000. – 1120 с.

3. Ролланд Ф. Д. Основные концепции баз данных // Ф. Д. Ролланд – М. : Вильямс, 2002. – 256 с.

4. Грофф Дж. Р. SQL: полное руководство // Дж. Р. Грофф, П. Н. Вайнберг, Э. Дж. Оппель – К. : Вильямс, 2015. – 959 с.

5. Кашеев Л. Б. Методичні вказівки до лабораторних робіт з курсу «Бази даних та інформаційні системи» для студентів напряму 6.040302 «Інформатика» / уклад. : Л. Б. Кашеев, С. В. Коваленко. – Х. : НТУ «ХПІ», 2015. – 56 с.